



腾讯云架构师技术同盟



Tencent Cloud
Valuable Professional

长风万里，剑犹在手

世易江湖改，道存功夫长

2026腾讯云西部架构师峰会

科技前沿，十年磨一剑

老江湖 VS 新生代，同台拆招

光明顶

缥缈峰

企业数智化，轻巧转型

妙辩吐槽技术江湖

少年游

青少年 AI 共创工坊

聚贤庄

Vibe Coding Jev LLM AI-Agent CloudNative LAMP www TCP
MCP Loop Engineering Linux
Transformer Harness iOS/Android
OpenClaw

长风万里，剑犹在手

世易江湖改，道存功夫长

→ 2026腾讯云西部架构师峰会 ←

基于具身机器人的AI模型应用

具身智能的模型趋势

主讲人：王一舟

Vibe Coding Jev LLM AI Agent
MCP Loop Engineering
Transformer Harness OpenClaw



王一舟

原力无限 CTO



<https://wangyizhou33.github.io/tech-blog/> (演讲材料)



wangyizhou3

01

行业

robotic vs driving vs digital AI

02

模型范式

VLA vs WAM

03

World model

renderer vs simulator vs planner

04

04

Training receipe

pre vs mid vs post training

05

数据类型

ego-centric vs teleop vs
synthetic

06

数采场

teleop vs intervention

07

model architecture
vs
training paradigm

04

08

finetuning vs
in-context learning

09

数据模态

vision vs tactile

10

RSI

data engine vs RL vs
autoresearch

11

Agentic 系统

code-as-policy vs harness

04

12

护城河

model vs AI infra

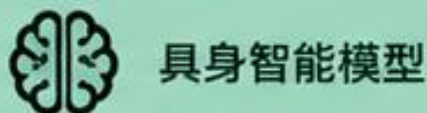


行业图谱：Robotic AI vs Driving AI vs Digital AI

Robotic AI



图像 + 关节角度 + 触觉 + 指令



关节控制指令



灵巧手控制指令



多模态感知与控制闭环，在物理世界中执行操作。

Driving AI



图像输入



自动驾驶模型

生成行驶轨迹



以视觉感知为主，生成轨迹并完成闭环驾驶。

Digital AI



Text Token
Prediction



Image Token
Prediction



多模态大模型



OpenAI

帮我写一个数据处理的
Python 脚本

```
python
import pandas as pd
df = pd.read_csv("data.csv")
...
```

文本 / 代码生成与理解

在数字世界中生成、理解与操作文本 / 图像 / 代码。



Seedance



图像 / 视频生成与理解



核心差异：Robotic AI 面向物理操作，Driving AI 面向轨迹决策，Digital AI 面向数字内容与软件环境。

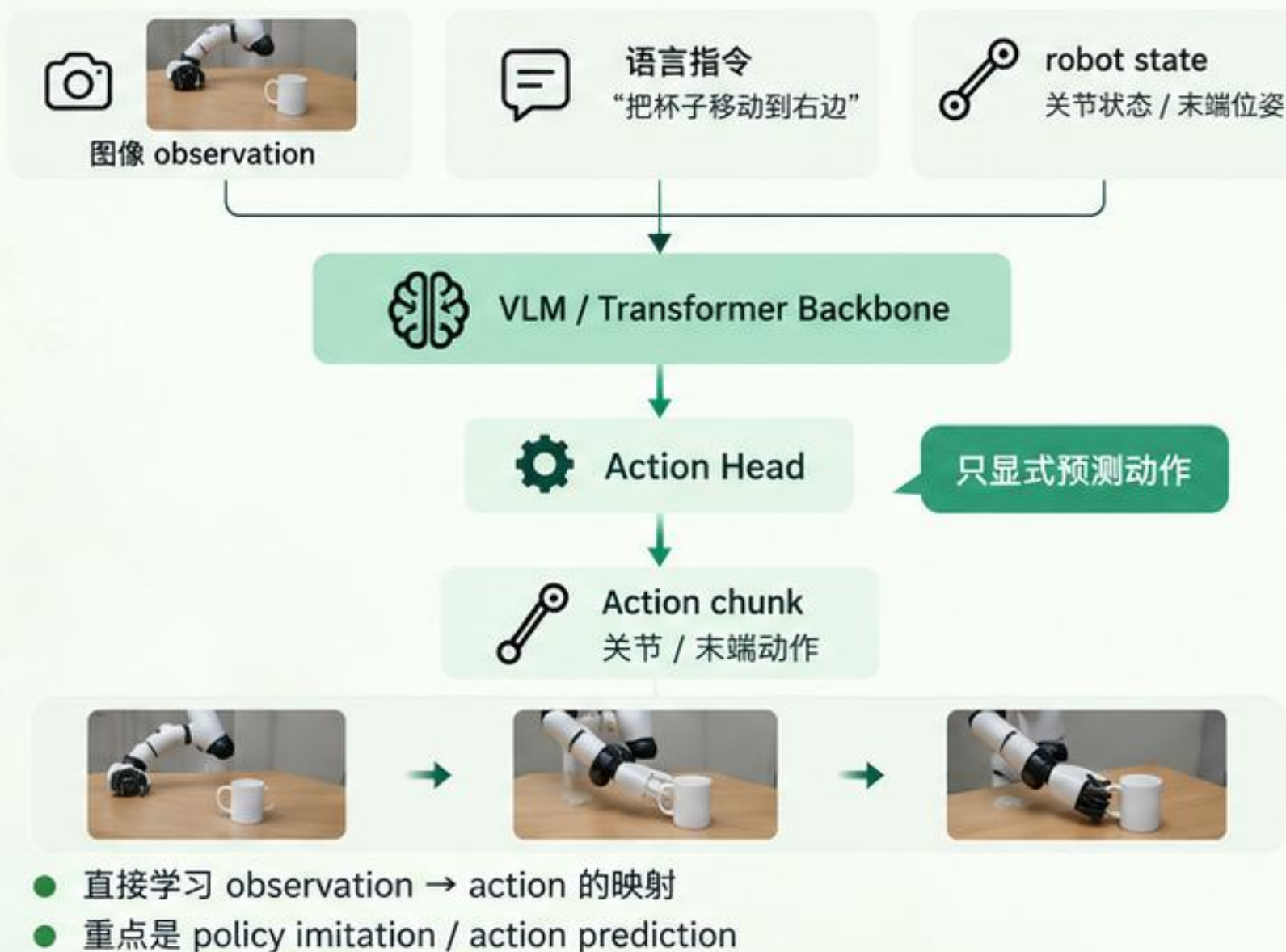


模型范式：VLA vs WAM

从 observation-to-action, 到 world-and-action modeling

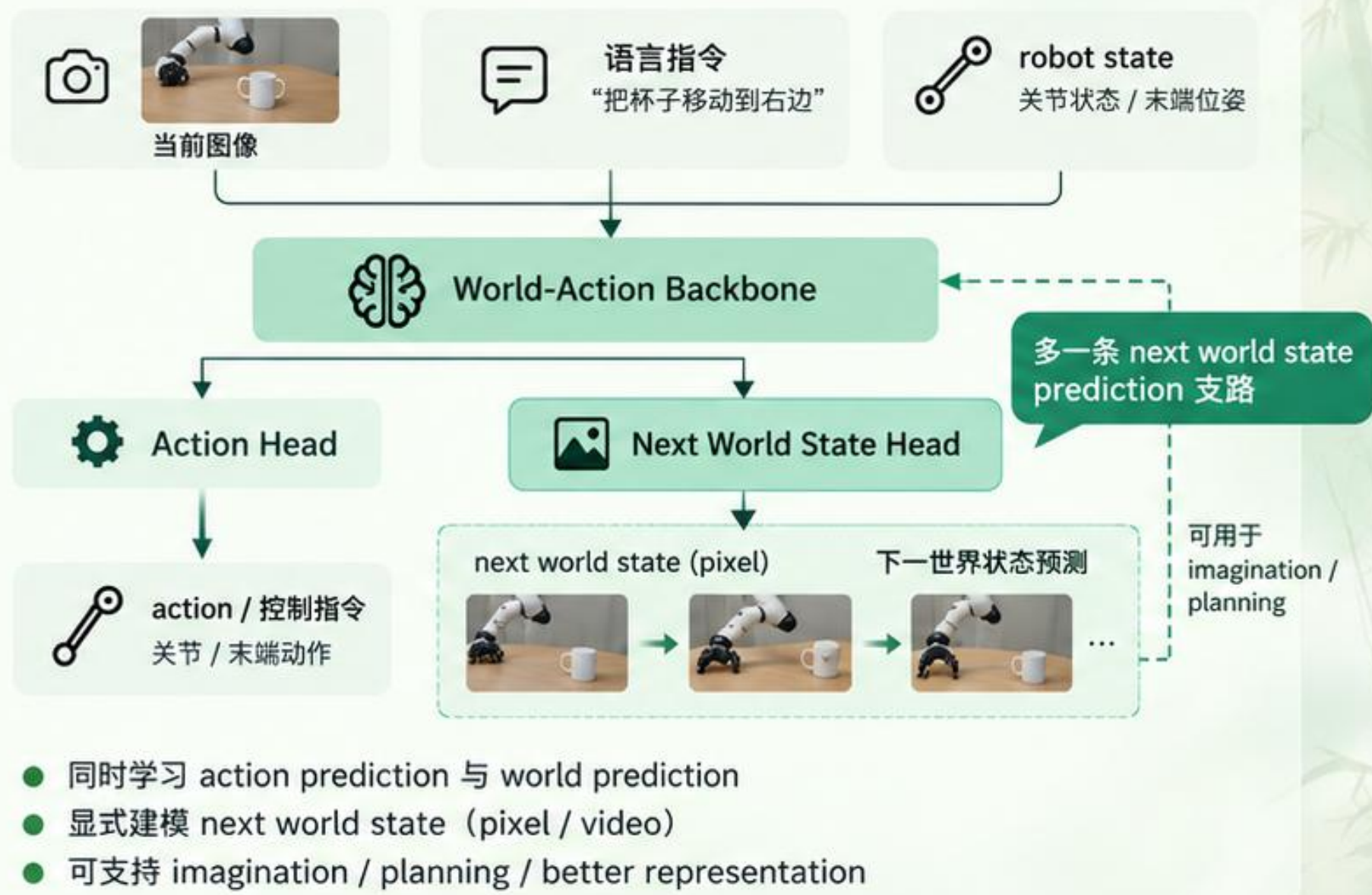
VLA

Vision-Language-Action



WAM

World-Action Model



核心差异：VLA 直接输出动作；WAM 既预测动作，也预测未来世界状态。



Takeaway: WAM 相比 VLA 的关键增量，不只是结构更复杂，而是把“世界如何变化”作为训练目标显式引入模型。



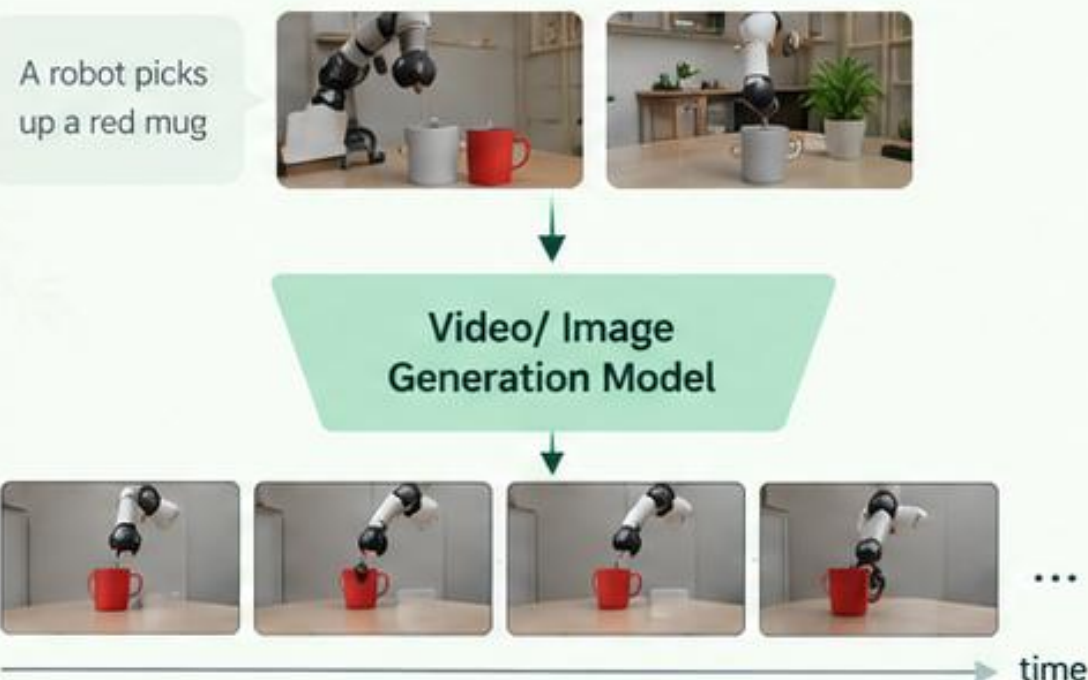
World Model 是一个被过度使用的概念

不同的世界模型在模型开发和应用中扮演完全不同的角色：生成逼真画面、帮助决策，或模拟物理演化

Renderer-like World Model

生成逼真的观测（像素）

从文本/图像/场景描述 生成高质量、一致的视觉内容



输入：文本 / 图像 / 场景描述



输出：高质量图像 / 视频



关注：视觉真实性、一致性、可控生成



典型应用：数据生成、仿真渲染、内容创作

Planner-like World Model

支持决策（从观测到动作）

理解场景、预测后果、选择最优行动



输入：当前观测（图像 / 状态）



输出：行动 / 轨迹



关注：任务完成、长期规划、决策效率

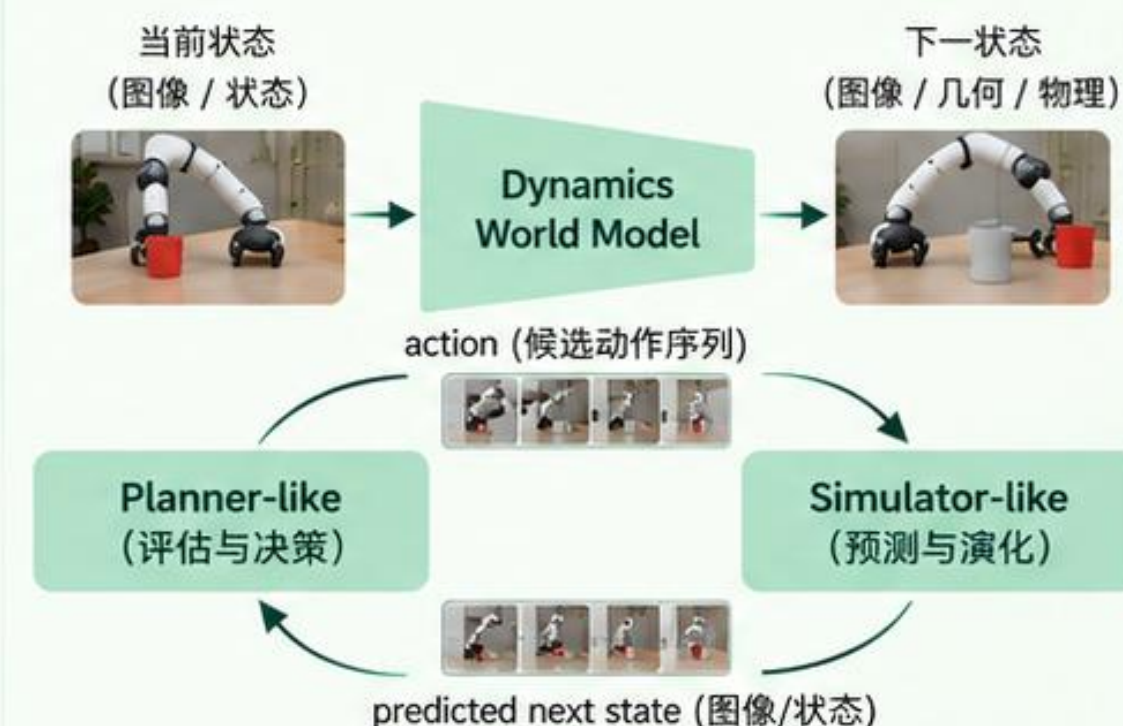


典型应用：任务规划、策略搜索、测试时推理

Simulator-like World Model

模拟物理世界的演化（从动作到下一状态）

在给定动作下，预测下一时刻的世界状态（像素 / 几何 / 物理）



输入：当前状态 + 动作



输出：下一时刻状态（像素 / 几何 / 物理）



关注：物理一致性、动力学准确性、可交互



典型应用：策略训练（MBRL）、数据合成、仿真评估



Takeaway:

World Model 并不是一个单一的技术路线，而是一个功能性的统称。根据应用目标不同，世界模型可以是用于生成的 Renderer、用于决策的 Planner，或用于预测物理演化的 Simulator。三者相互补充，共同支持具身智能系统的开发与应用。



训练范式：Pretraining vs Midtraining vs Posttraining

从通用先验，到具身操作，再到真实世界持续优化。



Takeaway

- Training recipe 不是单一阶段，而是分阶段吸收不同数据与反馈：Pretraining 学通用先验，Midtraining 学 manipulation 对齐，Posttraining 学真实世界中的持续优化。



从遥操为主到干预为主：数采场正转型为“脑训练场”

随着 Egocentric 数据提供了大规模、多场景的基础操作数据，数采场的定位从“采数据”转向“教模型在真实世界中更强地学习”。

遥操数据 Teleoperation Data

从人类示范中学习完成任务



- 人类远程操控机器人
- 提供完整的任务轨迹
- 高质量、可落地的示范数据
- 数据量有限、成本较高

干预数据 Intervention Data

在自主运行中纠正错误、处理长尾



- 模型自主运行，人在失效时介入
- 覆盖长尾场景与失败案例
- 真实世界分布，数据效率高
- 更适合规模化采集

从数采场



到脑训练场

以遥操采集为主
获取任务示范数据



以自主运行 + 干预为主
持续产生更有价值的数据



- ✓ 结合 Egocentric 大规模数据，减少纯遥操需求
- ✓ 更多用于模型自主运行、干预纠错、长尾探索
- ✓ 持续产出高价值数据，驱动模型真实世界学习
- ✓ 数采场升级为“脑训练场”，成为模型能力迭代的核心基础设施



Takeaway

- Egocentric 数据解决了数据量问题，干预数据解决了真实世界的难题。
- 数采场不再只是“收集数据”，而是“训练大脑”——让模型在真实世界中持续学习、自我改进。



Synthetic Data: 生成可控、多视角、物理一致的训练数据

Ego-centric 与 teleop 数据已在前文介绍；这里聚焦 synthetic data，尤其是 Atlas 的做法。

黑客帝国：拍摄手法



黑客帝国：成片效果



Atlas：拍摄手法



Atlas：成片效果



过去：生成与重建，两条互不搭界的路线



生成 (Generation)

- 凭想象画
- 今天多数视频生成模型主要走扩散路线。



重建 (Reconstruction)

- 照着拍到的东西还原
- 典型方法：NeRF / 3DGS。



Atlas：统一生成与重建的世界模型

Atlas 同时治好了两个一直被分开对待的毛病：把 generation 与 reconstruction 统一到一个可微、可控、可物理交互的世界模型里。

多视角视频 / bullet-time 采集



Atlas World Model

生成 + 重建

几何一致

物理一致

可控的 3D 场景与动作视频



Atlas 生成的 synthetic data 示例



Takeaway

● Synthetic data 的关键不只是“生成更多视频”，而是用世界模型同时获得几何一致、物理一致、可控可交互的数据。Atlas 给出了这条路线的代表性做法。

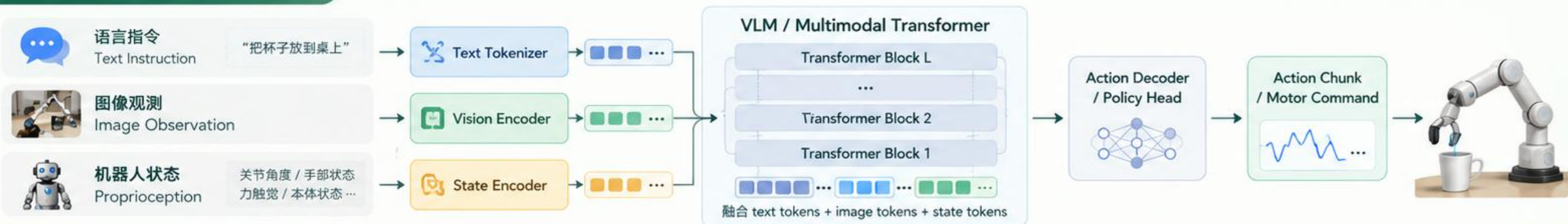


模型架构 vs 训练范式

同一个模型架构（如 VLA/Transformer），可以采用不同的训练范式（如行为克隆、强化学习等），模型架构定义“模型是什么”，训练范式定义“如何更新权重”。

模型架构 (Architecture)

定义模型的输入、模块组成、信息流与输出形式。



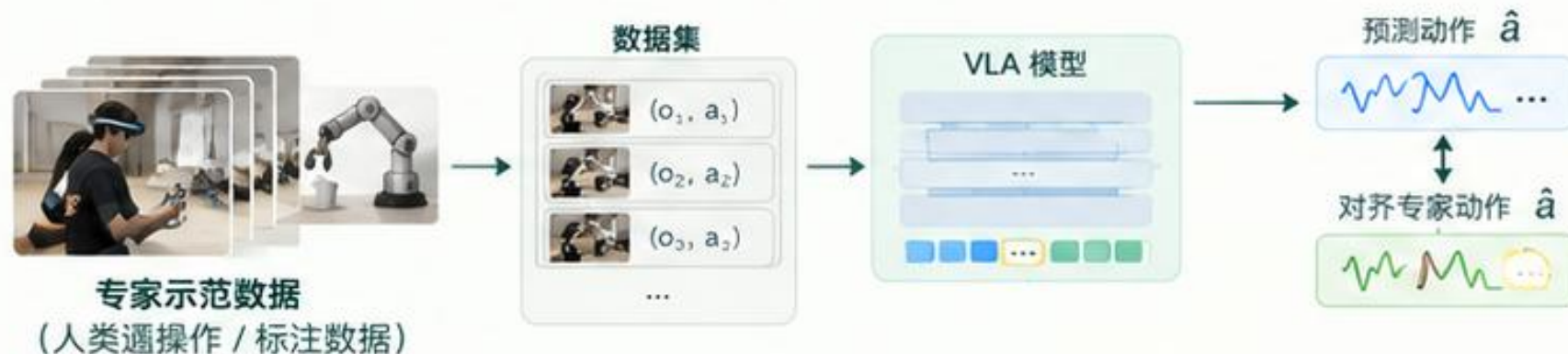
同一个 VLA 架构，可以用不同方式训练

训练范式 (Training Paradigm)

定义如何更新模型权重，即用什么信号、什么目标来优化模型。

1 监督学习 / 行为克隆 Behavior Cloning (BC)

在专家示范的数据上，学习“在这个状态下，应该做什么动作”。



2 强化学习 Reinforcement Learning (RL)

在真实或仿真环境中，通过试错与奖励信号，学习“什么动作能带来更好的长期回报”。



Takeaway

- 模型架构决定了模型“是什么”——可以是 VLA、WAM 等不同的结构，但都可以适配多种训练范式。
- 训练范式决定了“如何更新权重”——监督学习利用专家示范，强化学习通过环境反馈持续改进。

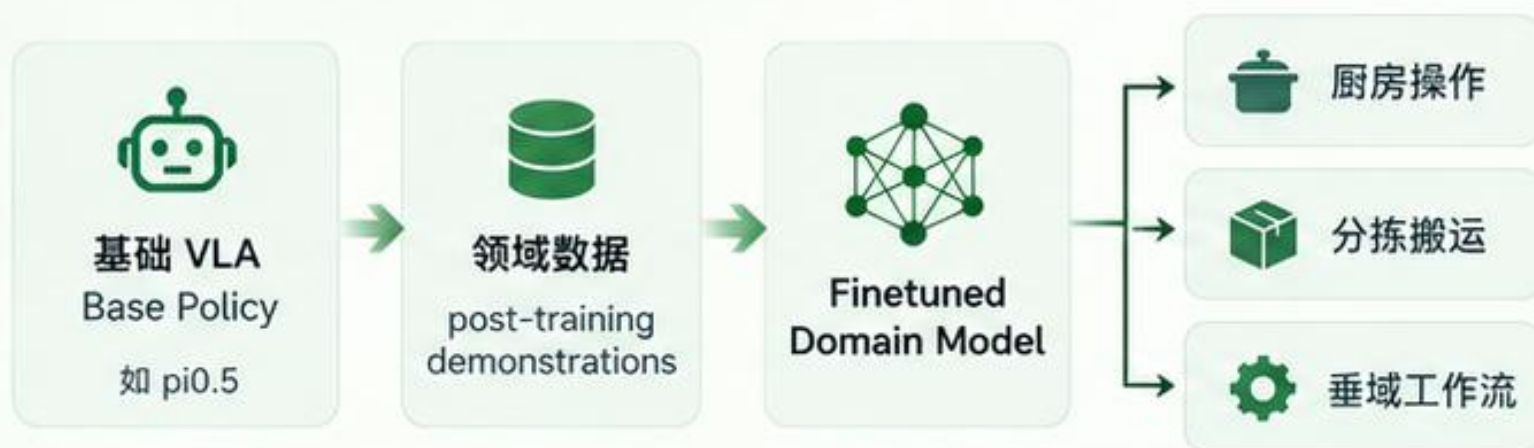


Finetuning vs In-Context Learning

任务适配的位置，正在从“模型权重”迁移到“上下文示范”。

Finetuning (权重适配)

以 pi0.5 及其衍生垂域应用模型为代表



- ✓ 任务知识主要写入模型权重
- ✓ 适配新任务通常需要新增示范 / finetuning
- ✓ 适合稳定、可重复的垂域落地



像 BERT 时代：通过改权重来适配任务

In-Context Learning (上下文适配)

以 Skild AI 与 Generalist Robotics 的 ICL 方法为代表



- ✓ 任务知识临时写入上下文，而非权重
- ✓ 可通过示范视频快速适配未见任务
- ✓ 更像 ChatGPT 时代：通过 prompt / context 适配任务

ICL 的样本效率



BERT → ChatGPT 的变化，在具身里对应：Finetuning → In-Context Learning



Takeaway: Finetuning 把任务写进权重，ICL 把任务写进上下文。对具身智能而言，ICL 有望像语言模型一样，提高新任务适配效率，并显著减少昂贵的 post-training / teleop 需求。



RSI: data engine vs policy RL vs auto-research

从数据闭环，到策略优化，再到 AI research capability 的递归增强

L1 Data-engine improvement

- 发现长尾 / 薄弱场景 → 采样、标注、回灌训练集 → 训练下一版模型。

典型：Tesla data engine



L2 Memory and skill improvement

- 不改模型权重，通过 memory / skill library / prompt / tool 使用经验累积能力。

典型：Agent memory



L3 Policy improvement through experience

- 策略自己与环境交互，采集 trajectory、reward、failure/success，再用 RL 更新 policy。

典型：Online RL



L4 Autonomous curriculum improvement

- 系统决定下一步学什么：选择任务、难度、场景，以及何时请求人类示范。

目标：最大化 learning progress



L5 Learning-process improvement

- 不只改 policy，而是改 training recipe: loss、reward、architecture、data mixture、optimizer、evaluator。

典型：AI for AI training



L6 Recursive AI-research improvement

- 更强的 AI researcher 设计出更强的 AI researcher，使改进 AI 的能力本身进入递归闭环。

完整的 RSI hypothesis



一个更清晰的 progression



L1 | Tesla Data Engine

- 找“缺什么数据”



L3 | RL

- 自己产生 experience



L4 | Curriculum Engine

- 决定“下一份最值得产生的 experience 本什么”



L5 | Learning-process improvement

- 决定“怎样改变 learning algorithm 本身”

数据闭环

(发现并获得更多更好的数据)

策略优化

(从 experience 中学会更好行为)

自主课程

(产生更有价值的 experience)

算法自我改进

(改进学习算法本身)

这条链路把 Data Engine、RL、Curriculum 与更深层的 RSI 区分开来。



模态：视觉 vs 触觉

当前模型输入的优先级大致是：视觉 > 语言 > 触觉

触觉同样重要，但在当前场景选择上，我们会避免那些特别依赖触觉反馈的应用。



1
视觉
(主输入)



2
语言
(辅助输入)



3
触觉
(场景化输入)



视觉模态（当前主流）

典型采集设备：相机 / 头显 / IMU / 夹爪侧传感器



- ✓ 输入最成熟，覆盖面广
- ✓ 适合大多数操作任务
- ✓ 因此当前落地以视觉为主



触觉模态（开始布局，但暂不作为优先场景）

典型采集设备：触觉手套、肌电手环



- ✓ 对精细接触与力反馈很重要
- ✓ 长期看是关键增量模态
- ✓ 但当前数据与模型仍有瓶颈，因此在场景选择上会规避强依赖触觉的应用



优先落地的场景（以视觉为主，触觉非必需）

商业服务



迎宾接待、引导问询
配送服务

工业制造



物料搬运、产线协作
质量检测

零售物流



货物拣选、仓储搬运
库存盘点

文旅互动



导览讲解、互动体验
文化传播

家庭陪伴



陪伴教育、生活助理
家务协作

教育科研



科研平台、教学实验
具身智能研究



暂时规避的场景（强依赖触觉反馈，数据/模型仍有瓶颈）

精细装配



精密插拔、柔性装配
需要精细力控

柔性操作



柔软物体抓取、折叠
需要触觉感知

复杂接触操控



易滑/不规则物体操作
需要实时力反馈



Takeaway

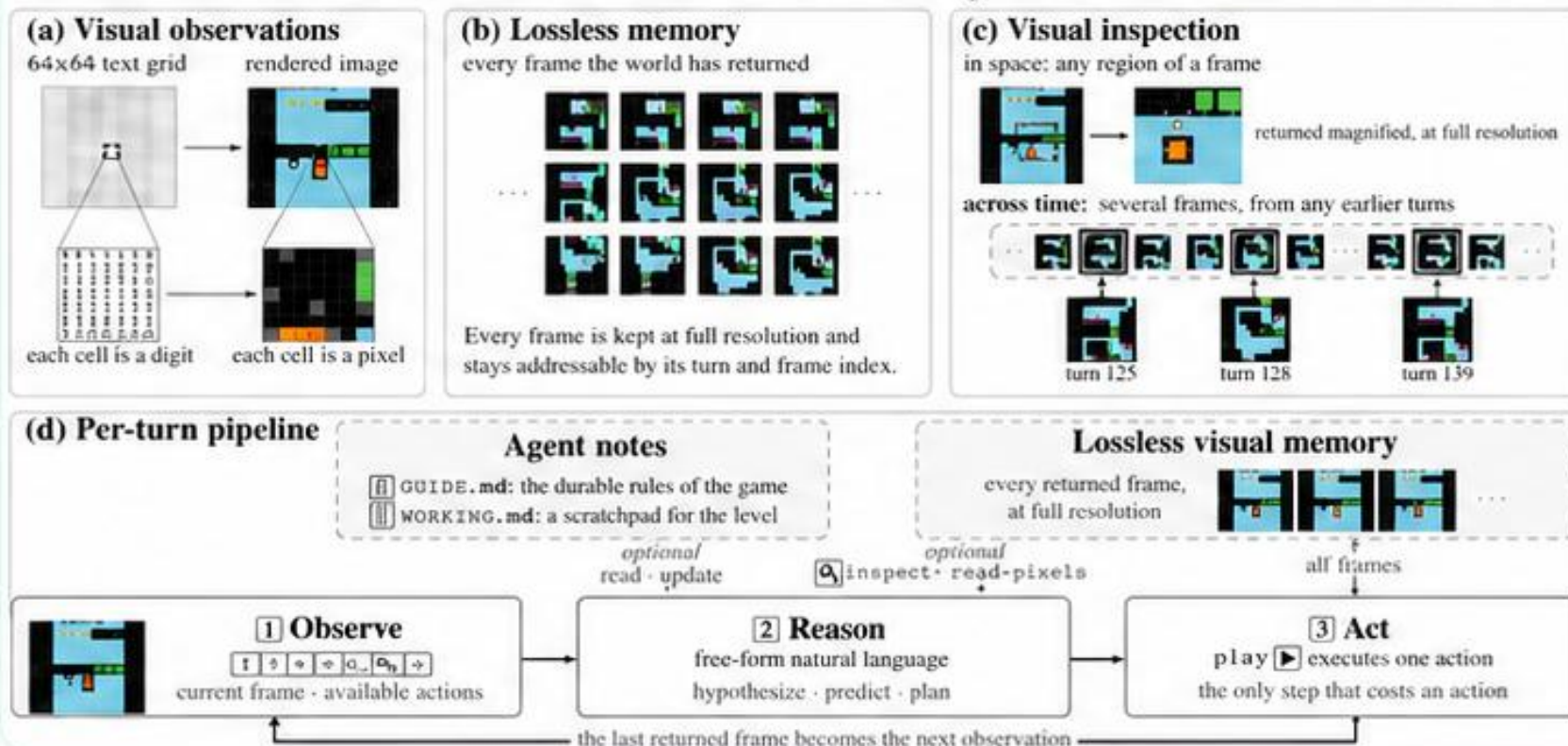
- 视觉仍是当前具身模型的主输入模态，能够支撑大多数通用场景的落地。
- 触觉很重要，但在数据与模型能力有限的阶段，我们会选择不强依赖触觉反馈的应用，优先落地视觉可支撑的场景。

LLM + Agentic 系统直接部署在具身智能

在 Robo-Coliseum 榜单上，多个 frontier LLM 直接出现在具身任务排名中，说明LLM也可以直接部署到具身系统。

Overall		Instruction Following		Spatial Reasoning		Robustness		General Manipulation		Protocol →
Rank	Account	Model	Total Score	Instruction Following	Spatial Reasoning	Robustness	General Manipulation	Open Source		
1	Primotion	Paimon	0.7754	0.8961	0.6791	0.8653	0.6613			
2	HiDream.ai	HiDream-O1-Embodied	0.7692	0.8913	0.7437	0.7978	0.6442			
3	Claude Opus 5.5	Claude Opus 5.5	0.7628	0.9660	0.8250	0.7930	0.4670			
4	GPT-6 Astra	GPT-6 Astra	0.7599	0.9935	0.7833	0.8042	0.4586			
5	GPT-6.1 Sol	GPT-6.1 Sol	0.7090	0.9550	0.7771	0.7040	0.4001			
6	Dexmal	DM0.5	0.7076	0.8444	0.6146	0.7344	0.6370	View		
7	Gemini 4 Argon	Gemini 4 Argon	0.6685	0.8589	0.6938	0.7336	0.3879			
8	GigaAI	GigaBrain-0.7	0.6504	0.8166	0.4729	0.6800	0.6320	View		

VISTA: A Visual Harness for Reasoning in an Interactive World



1 Code-as-Policy

LLM 主要扮演 coding agent 的角色



- LLM 负责把任务翻译成可执行代码
- 代码中显式调用 tool / robot API
- 适合结构清晰、接口稳定的任务

2 Harness

由 Agent 系统根据模型 response 来调用 tool



- 模型给出意图、步骤或中间响应
- Agent 系统负责工具调用与上下文管理
- 适合复杂流程、长任务与多技能协同



Takeaway: frontier LLM 进入具身榜单，并不意味着都要训练成专用 policy。现实中常见的两条路径是：Code-as-Policy (LLM→code→toolcall) 和 Harness (LLM response→agent system→toolcall)。



小脑平台 vs 大脑平台

两类平台分别面向动作技能生产与通用任务智能生产，但都服务于模型到本体/场景的持续部署。



小脑平台

动作视频 → 小脑模型 → 动作编排 → 本体 OTA

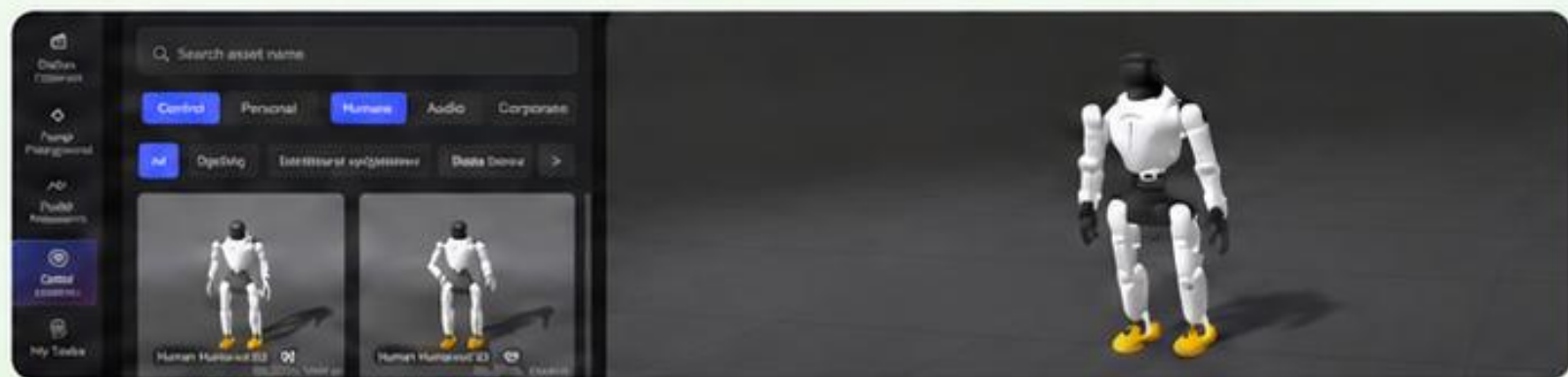
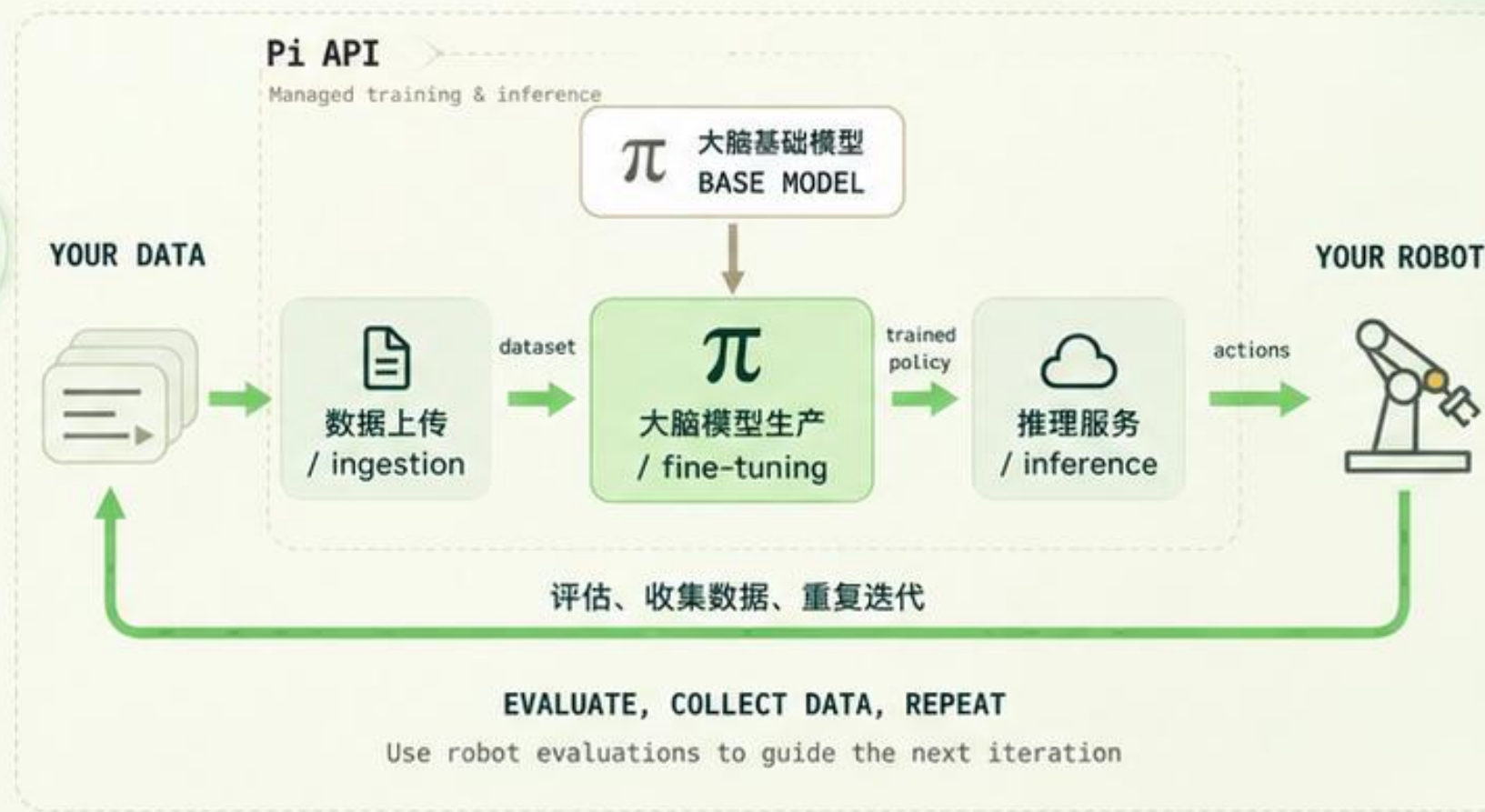


VS



大脑平台

数据上传 → 大脑模型 → 场景部署



- 面向 locomotion / manipulation 等动作技能
- 强调 motion imitation、技能复用与编排
- 最终部署到具体本体，形成可执行的小脑能力



Takeaway

行业小脑平台已经成熟，且门槛较低。大脑平台由 physical intelligence 跨出第一步，给了人想象空间



具身智能的竞争壁垒

模型能力 vs AI Infra 能力

大脑公司一定是模型能力和 AI Infra 能力螺旋上升的，两者相互强化，缺一不可。



模型能力

上限高，但不确定性大

通过更好的模型架构、训练方法和数据配方，持续提升通用泛化能力与任务成功率。



操作技能



长程规划



泛化适应



人机交互

短期视角

- 上限高：一旦突破，可以带来显著的性能提升与更广泛的应用场景。
- 不确定性大：新模型架构、新训练方法的出现，可能快速颠覆现有优势。
- 需要持续探索：例如 VLA、世界模型、记忆、RSI 等仍在快速演进。

更强的
具身智能系统

更好的数据

更强的模型

更高的应用价值



AI Infra 能力

确定性强，支撑持续迭代

覆盖数据和模型的全生命周期管理，构建可持续的数据飞轮和工程化体系，支撑模型能力的快速迭代与规模化落地。



数据采集

多模态
多场景



质检清洗

质量门
可追溯



标注处理

自动化
人机协同



训练

分布式
高效训练



评估

仿真 + 真机
自动化评测



回流

失败样本
新场景挖掘

短期视角

- 确定性强：无论模型路线如何演进，数据与工程化基础设施始终是必需的。
- 可持续积累：随着数据规模、质量和工具链的完善，壁垒逐步增强。
- 应对未知迭代：能够快速适配新的模型方法、数据需求和应用场景。



Takeaway

- 模型能力决定了我们的上限，AI Infra 能力决定了我们能否持续快速迭代并落地。
- 两者相互强化、螺旋上升，是大脑公司的核心壁垒。



腾讯云架构师技术同盟



Tencent Cloud
Valuable Professional

谢谢观看

THANKS

